**CASE STUDY:**

**HERITAGE BITES RESTAURANT**

***by***

***NITIN***

***Southern Alberta Institute of Technology* Calgary, Canada**

**https://www.linkedin.com/in/nitin-b57867168/**

Contents

[**CASE STUDY:**](#_Toc178976477) [**HERITAGE BITES RESTAURANT**](#_Toc178976478)

[**1. Introduction to Heritage Bites Restaurant and its Database Design** 2](#_Toc178976483)

[**1.1 Purpose of the Database Design** 2](#_Toc178976484)

[**1.2 Overview of Heritage Bites Restaurant** 2](#_Toc178976486)

[**2. Mission & Objective** 2](#_Toc178976488)

[**2.1 Mission Statement** 2](#_Toc178976489)

[**2.2 Business Objectives** 2](#_Toc178976490)

[**3. Database Design** 3](#_Toc178976491)

**4. Table Relationship**…...……………………………………………………………………………3

**5. Entity Relationship Diagram (ERD)** ……………...……………………………………………4

**6. Conclusion** ………………………...………………………………………………………………4

**7. Appendix** ................……………………………………………………………………………5-11

[**7.1 Appendix A** 5-8](#_Toc178976505)

[**7.2 Appendix B** 8-11](#_Toc178976505)

**1. Introduction to Heritage Bites Restaurant and its Database Design**

**1.1 Purpose of the Database Design**

Heritage Bites Restaurant wants to manage its operations better; therefore, it needs an effective database system design to enable easy implementation of basic functions such as customer management, reservations, order taking, and inventory. With only 8 tables, the challenge remains in how to enhance efficiency without losing the quality of customer satisfaction.

**1.2 Overview of Heritage Bites Restaurant**

Heritage Bites is a homely restaurant that uses only locally sourced raw materials, adding a personal touch to its service. Restaurants with a limited number of tables will have to run every operational aspect of reserving to maintaining an inventory, just right. This will enable the restaurant to function efficiently sans any compromise on quality with the help of a well-structured database.

**2. Mission & Objective**

**2.1 Mission Statement**

To be the leading Southeast ethnic restaurant, ensuring the delivery of a delightful dining experience.

**2.2 Business Objectives**

1. **Efficient Table Management**  
   Minimize wait times and maximize seating utilization to ensure a smooth dining experience.
2. **Order Tracking**  
   Ensuring timely preparation and delivery of food.
3. **Inventory Management**  
   Maintain stock for menu items and reduce waste, ensuring that ingredients are always fresh and available.
4. **Customer Data**  
   Personalize services to enhance the dining experience based on customer preferences and order history.

**3. Database Design**

The restaurant's database will consist of several key entities, each serving a specific purpose:

1. **Customers**
2. **Employees**
3. **Reservations**
4. **Tables Table**
5. **Menu\_Items**
6. **Orders**
7. **Order\_Details**
8. **Payments**

4. **Table Relationship**

Here’s a brief explanation of the relationships between the tables in the restaurant database:

1. **Customers → Reservations** (One-to-Many)

A customer can make multiple reservations, but each reservation belongs to one customer.

1. **Tables → Reservations** (One-to-Many)

A table can be reserved many times, but each reservation is for one specific table.

1. **Employees → Orders** (One-to-Many)

An employee can handle multiple orders, but each order is managed by one employee.

1. **Customers → Orders** (One-to-Many)

A customer can place multiple orders, but each order is tied to a single customer.

1. **Tables → Orders** (One-to-Many)

A table can have multiple orders over time, but each order is associated with one table.

1. **Orders → Order\_Details** (One-to-Many)

An order can contain multiple items (order details), but each detail belongs to one order.

1. **Menu\_Items → Order\_Details** (One-to-Many)

A menu item can appear in multiple order details, but each order detail is for one menu item.

1. **Orders → Payments** (One-to-One)

Each order has exactly one payment, and each payment is linked to one order.

**5. Entity Relationship Diagram (ERD)**



**6. Conclusion**

The proposed structure of the database aids in effective management at restaurants for better customer satisfaction, order processing, and inventory management. In other words, this could be the backbone for managing data to ensure that restaurants can provide a delightful dining experience.

**7. Appendix**

**7.1 Appendix A**

1. **Customer Table**

**Description:** Stores customer details for contact and loyalty tracking. Used for managing customer interactions and personalized services.

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| Customer\_ID | INT | Unique identifier for each customer (Primary Key) |
| First\_Name | VARCHAR(50) | Customer's first name |
| Last\_Name | VARCHAR(50) | Customer's last name |
| Phone | VARCHAR(15) | Customer's phone number |
| Email | VARCHAR(100) | Customer's email address |
| Loyalty\_Points | INT | Points earned by the customer for loyalty programs |

1. **Employees Table**

**Description:** Holds employee data, including role and salary. Helps with shift scheduling, payroll, and staff management

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| Employee\_ID | INT | Unique identifier for each employee (Primary Key) |
| First\_Name | VARCHAR(50) | Employee's first name |
| Last\_Name | VARCHAR(50) | Employee's last name |
| Role | VARCHAR(50) | Employee's job role |
| Shift\_Time | VARCHAR(20) | Employee's working shift |
| Salary | DECIMAL(8,2) | Employee's salary |

1. **Reservations Table**

**Description:** Manages reservation information, linking customers to tables. Tracks reservation status (confirmed, pending) for efficient table use.

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| Reservation\_ID | INT | Unique identifier for each reservation (Primary Key) |
| Customer\_ID | INT | Foreign key linking to the customer who made the reservation |
| Table\_ID | INT | Foreign key linking to the table being reserved |
| Reservation\_Time | DATETIME | Time of the reservation |
| No\_of\_Guests | INT | Number of guests for the reservation |
| Status | VARCHAR(20) | Reservation status (Confirmed or Pending) |

1. **Tables Table**

**Description:** Details of table capacities and locations within the restaurant. Helps in managing seating and reservations efficiently.

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| Table ID | INT (PRIMARY KEY) | A unique identifier for each table, ensuring no duplicate entries. This serves as the primary key for the table. |
| Capacity | INT | The number of guests each table can accommodate helps in assigning appropriate tables during reservations. |
| Location (e.g., Indoor, outdoor) | VARCHAR (10) | Specifies the table’s location, such as "Indoor" or "Outdoor," assisting staff in seating customers based on their preferences or availability. |

1. **Menu\_Items Table**

**Description**: Stores details of food and drinks offered on the menu. Helps in managing what is available for customers to order.

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| Item\_ID | INT | Unique identifier for each menu item (Primary Key) |
| Name | VARCHAR(100) | Name of the menu item |
| Category | VARCHAR(50) | Category of the menu item (e.g., Main Course, Dessert) |
| Price | DECIMAL(5,2) | Price of the menu item |
| Description | TEXT | Description of the menu item |

1. **Orders Table**

**Description**: Tracks customer orders, including which employee took the order. Used to manage the flow of orders during service hours.

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| Order\_ID | INT | Unique identifier for each order (Primary Key) |
| Customer\_ID | INT | Foreign key linking to the customer who placed the order |
| Employee\_ID | INT | Foreign key linking to the employee who took the order |
| Table\_ID | INT | Foreign key linking to the table associated with the order |
| Order\_Time | DATETIME | Time the order was placed |

1. **Order\_Details Table**

**Description**: Contains detailed information about each order’s contents. Helps to track the specific items ordered and the quantities.

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| Order\_Detail\_ID | INT | Unique identifier for each order detail (Primary Key) |
| Order\_ID | INT | Foreign key linking to the corresponding order |
| Item\_ID | INT | Foreign key linking to the menu item ordered |
| Quantity | INT | Quantity of the item ordered |

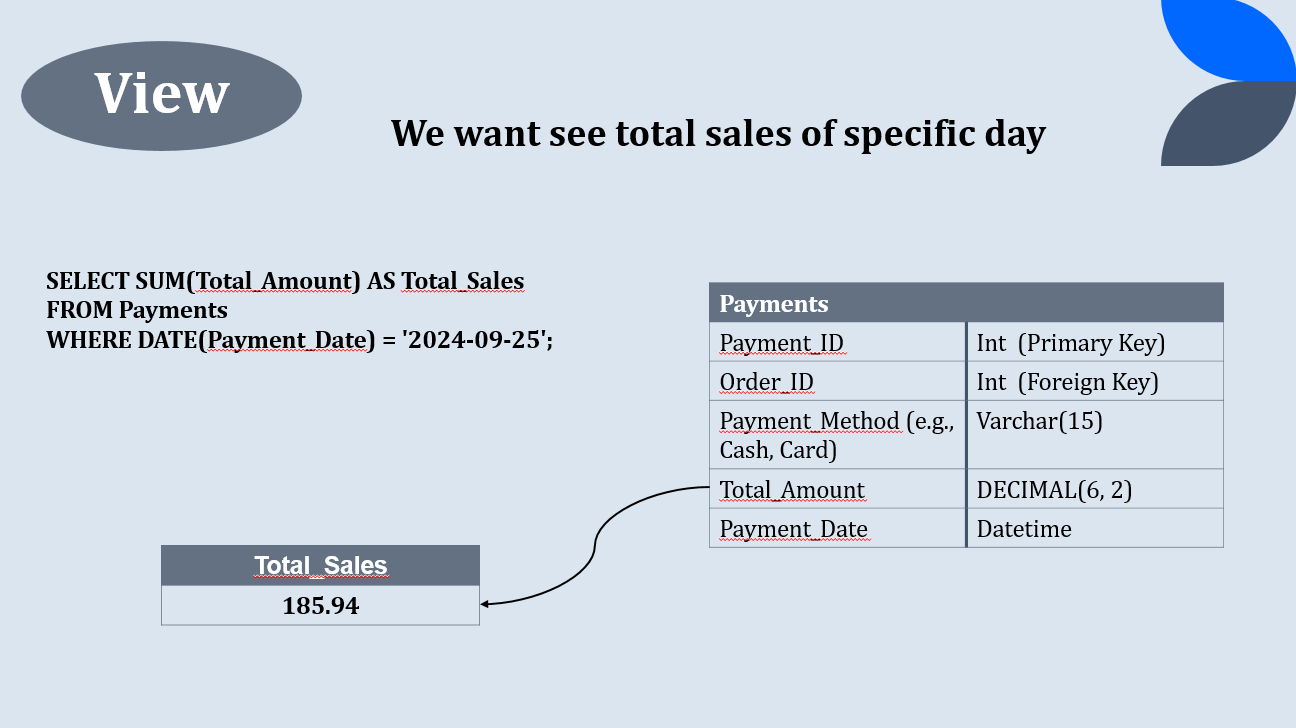
1. **Payments Table**

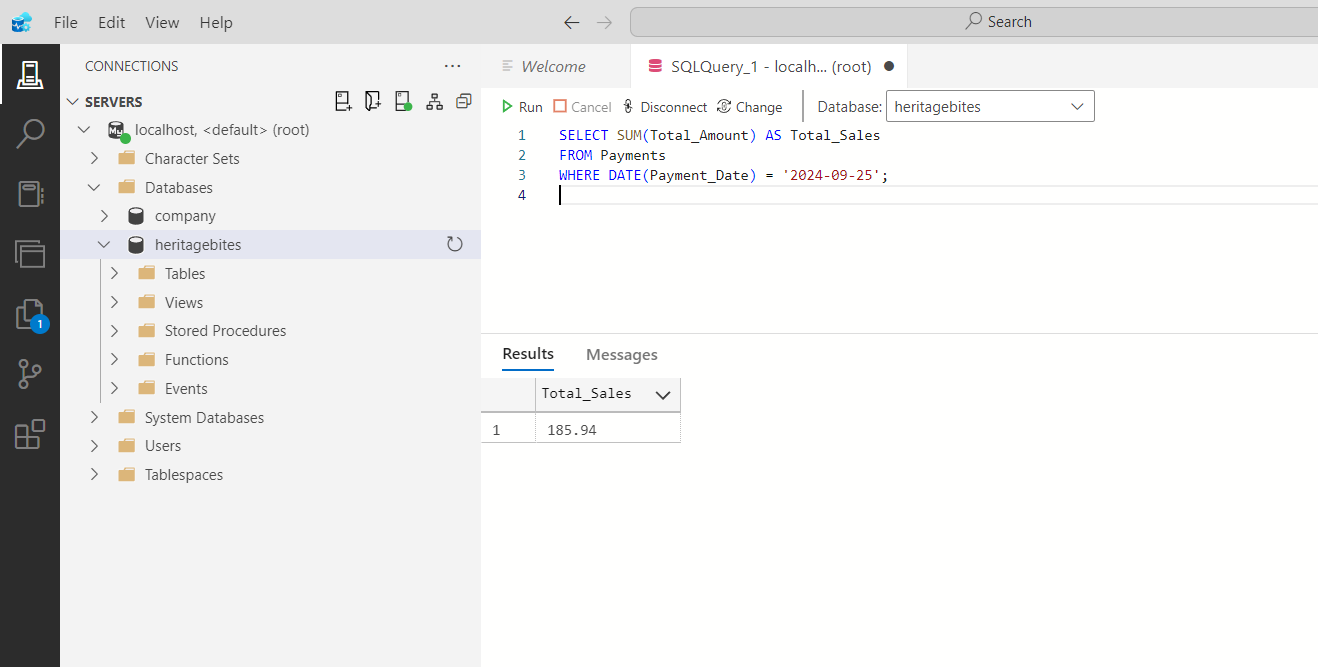
**Description:** Stores payment transaction details like method and amount. Tracks how orders are paid and manages financial records.

|  |  |  |
| --- | --- | --- |
| Field | Data Type | Description |
| Payment\_ID | INT | Unique identifier for each payment (Primary Key) |
| Order\_ID | INT | Foreign key linking to the order for which payment is made |
| Payment\_Method | VARCHAR(20) | Method of payment (e.g., Credit Card, Cash) |
| Total\_Amount | DECIMAL(8,2) | Total amount paid for the order |
| Payment\_Date | DATETIME | Date and time of payment |

**7.2 Appendix B: Queries**

* **Testing Database and Query**





* **Join Query**

